MIG at the NTCIR-15 FinNum-2 Task: Use the Transfer Learning and Feature Engineering for Numeral Attachment Task

Yu-Yu Chen

FinNum-2 Task

The goal is to judge whether the specified numeral is related to the given stock symbol in a financial tweet.

“The **AAPL** stock price increases **30%** in a few days”

Data Format Example

"**tweet**": "$SQ is $39 per share with a P/E of over 150 and losing money...should be at least as good as them with a P/E of 30 and making money!!",

"**target_num**": "39",

"**target_cashtag**": "SQ",

"**relation**": 1,

"**offset**": 8

Introduction

We use 20% as test set, 10% as development set, and 70% as train set.

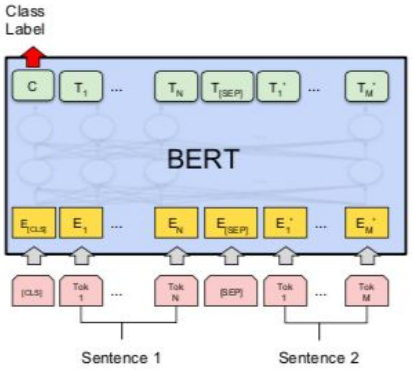
We further take apart the train set to the validation set and the train set.

In our research, we fine-tuned the BERT and applied linguistic domain knowledge into new feature.

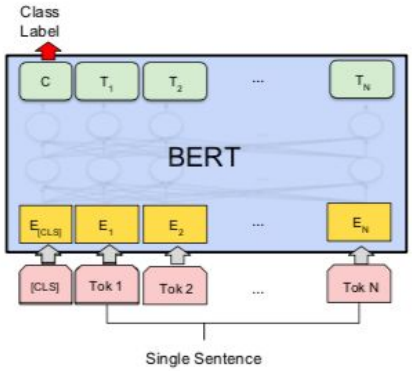
Method: Fine-tune BERT

BERT model can fit into several tasks such as

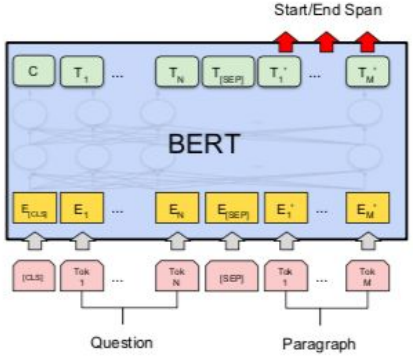
- sentence classification
- sentence pairing
- question answering
- sentence tagging.



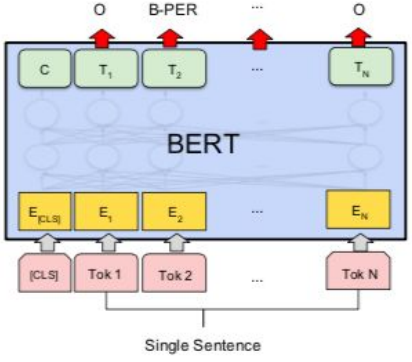
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Method: Fine-tune BERT

We use BERT-large as our pre-trained model because the BERT-large performs better than BERT-base model in our task.

Table 8: BERT-base vs. BERT-large

	Macro-F1
BERT-base	0.8488
BERT-large	0.8823

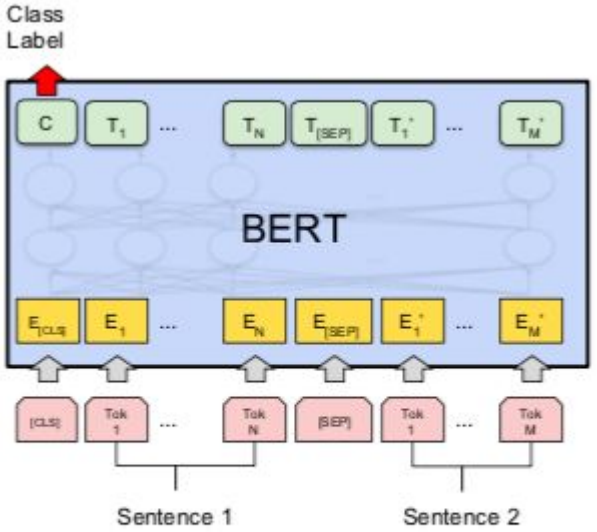
Method: Fine-tune BERT

Here we treat our task as sentence pairing and we use features as the second sentence.

Hypothesis:

The text embeddings we learned can capture the main idea of a tweet.

If the target cashtag and target number are correlated, they will also relates to the meaning of tweet



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

Method: Fine-tune BERT

Since there is data unbalance in our data, we set the weight of loss function to response to this situation.

In all tweets, about 82% of the cashtags and numbers in tweets has a relation. Therefore we try some different weights to have better results.

We also try to add a Bi-directional Long Short-Term Memory(BiLSTM) after the layer of BERT.

And we tuned the number of epoch to reach a better performance.

Method: Feature Engineering-1&2

The features we found useful is related to the cashtag and number.

Assumption:

In English, the main idea are usually in the first sentence, which is called topic sentence. So if a word has an important meaning in a sentence, it will be put in front.

We did the word segmentation and numbered the word in order of the position in a tweet.

And the numbers we generated are the **position features**.

Method: Feature Engineering-1&2

For example, we encode this sentence:

“$NE, last time oil was over $65 you were close to $8.”

0 1 2 3 4 5 6 7 8 9 10 11

The cashtag is $NE, so we generate a position feature of 0.

The number is $65, so we generate a position feature of 6.

Method: Feature Engineering-3

Assumption:

In the view of linguistics, if the distance between words are close, it might has higher probability that the words are correlated.

Usually a noun comes with an adjective or a verb.

In our case, the cashtag and the number are both Noun so the patterns might be N.+V.+N. or N.+Adj.+N. or other patterns.

As the result, we treat the distance between the cashtag and the number as the **distance feature**.

Experiment

Because most of our data are labeled as 1, we guess all the data 1 and treat it as the baseline model(Here we called null model).

Then we try to use the features that are in the **original** form of data, which includes the offset of the target number and the number itself.

We use these two features to train a multiple layer perceptron(MLP).

Table 5: Null model and numeral features in MLP

	Macro-F1
Null model	0.4872
Original features	0.4811
New features	0.7600

Experiment

By using the three syntactic features:

- the position of cashtag
- the position of number
- the distance between cashtag and number

The performance significantly increases.

Table 5: Null model and numeral features in MLP

	Macro-F1
Null model	0.4872
Original features	0.4811
New features	0.7600

Method: Feature Engineering-Average

Assumption:

Some of tweets have several sentences. The distance might be high and the effect will be diluted.

Therefore, we need to average them with the length of entire tweet. We create three more features which include average position of cashtag, average position of number and average distance between them.

Totally we create six new features.

Experiment Results

Table 7: Evaluation of the experiments

	Macro-F1
Single sentence	0.8488
Words features	0.8457
3 new features	0.8571
6 new features	0.8636
7 new features	0.8618

Other Features-Words Features

We also try to use the word of cashtag and number themselves as input. We treat the two words as sentence two for the sentence pair task in BERT.

The performance seems not sensitive to the two words.

It’s just slightly worse than a single sentence.

Other Features-Keyword Features

We extract the related data and find some keywords by TF-IDF analysis.

We create a dictionary:

['will', 'up', 'amp', 'shares', 'more', 'today', 'now', 'over', 'down', 'buy', 'next', 'stock', 'out', 'day', 'back', 'just', 'target', 'short', 'bought', 'price', 'get', 'year', 'last', 'week', 'SPY', 'market', 'million', 'BTC', 'close', 'calls', 'trading', 'AMD', 'sell', 'bullish', 'big']

Settings of Formal Runs

Run1:

loss function weight 0.99 and 0.01 / additional BiLSTM layer / 5 epochs

Run2:

loss function weight 0.8 and 0.2 / additional BiLSTM layer / 8 epochs

Run3:

loss function weight to 0.9 and 0.1 / without additional BiLSTM layer / 8 epochs

Results

In Test Set:

Run2 > Run3 > Run1

In Development Set:

Run3 > Run2 > Run1

In Validation Set:

Run2 > Run1 > Run3

Table 5: Experimental results. (%)

Team	Development	Test
Majority	44.88	44.93
CYUT-1	48.64	48.02
WUST	82.91	54.43
BTBCH-1	100.00	57.19
BTBCH-2	99.68	58.00
3edc-3	87.34	58.40
3edc-2	85.17	59.77
IIITH-1	96.16	62.81
Caps-m [2]	79.27	63.37
IIITH-3	93.99	64.16
3edc-1	87.02	64.76
MIG-1	84.46	68.27
MIG-3	90.69	68.37
TLR-2	87.81	68.64
MIG-2	85.77	68.72
IIITH-2	96.23	71.11
TLR-1	88.26	71.41
CYUT-2	95.99	71.90
TLR-3	88.87	73.95

Table 2: Results of Run 1

# of epochs	Val set	Dev set
5	0.8021	0.8446
8	0.8251	0.8354
12	0.8531	0.8178
15	0.8585	0.7786

Table 3: Results of Run 2

# of epochs	Val set	Dev set
5	0.8542	0.7845
8	0.8473	0.8577
12	0.8573	0.8044

Table 4: Results of Run 3

# of epochs	Val set	Dev set
5	0.8573	0.7851
8	0.7252	0.9069

Conclusions

In the research, we find that linguistic concepts can help computers to recognize the relationship between entities.

The result represents that the knowledge of linguistics is important to the relation recognition task.

Nowadays we have high-dimensional pre-trained embeddings, so we don’t have to spend a lot of time to train the text embeddings whereas we can achieve the basic meaning in a text.

Future Work

In the future, we can focus more on the relationship of the domain keywords, the cashtag and the number. Because in the field of finance, the financial keywords usually have an important meaning in the text.

Also, we can study more on unrelated data and extract some features that can represent them.

Thank you